QBUS6850

Lecture 9

Neural Network and Deep Learning- I

BUSINESS SCHOOL

QBUS6850 Team

THE UNIVERSITY OF SYDNEY

- Topics covered
  - Neural network (NN) intuition
  - Neural network representation
  - Forward propagation
  - Neural network examples: Boolean functions

- References
  - Alpaydin (2014), Chapter 11
  - Bishop (2006), Chapter 5
  - https://am207.github.io/2017/wiki/gradientdescent.html#stochastic-gradient-descent

Learning Objectives

- Understand the intuition of NN
- Understand the how NN can be used for regression and classification
- Understand the how forward propagation NN works
- Understand how NN can be applied to realize Boolean functions

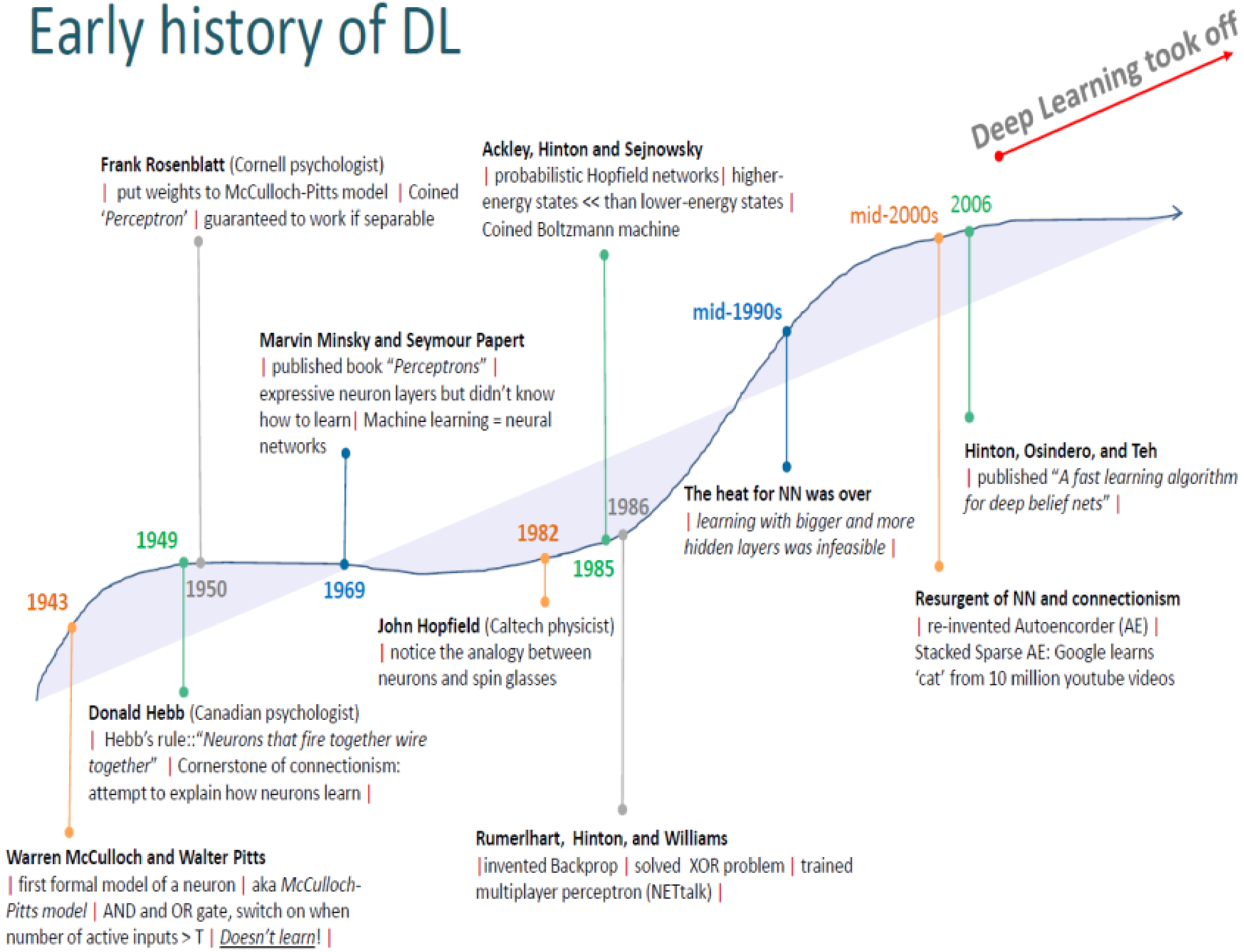
Neural Network Intuition

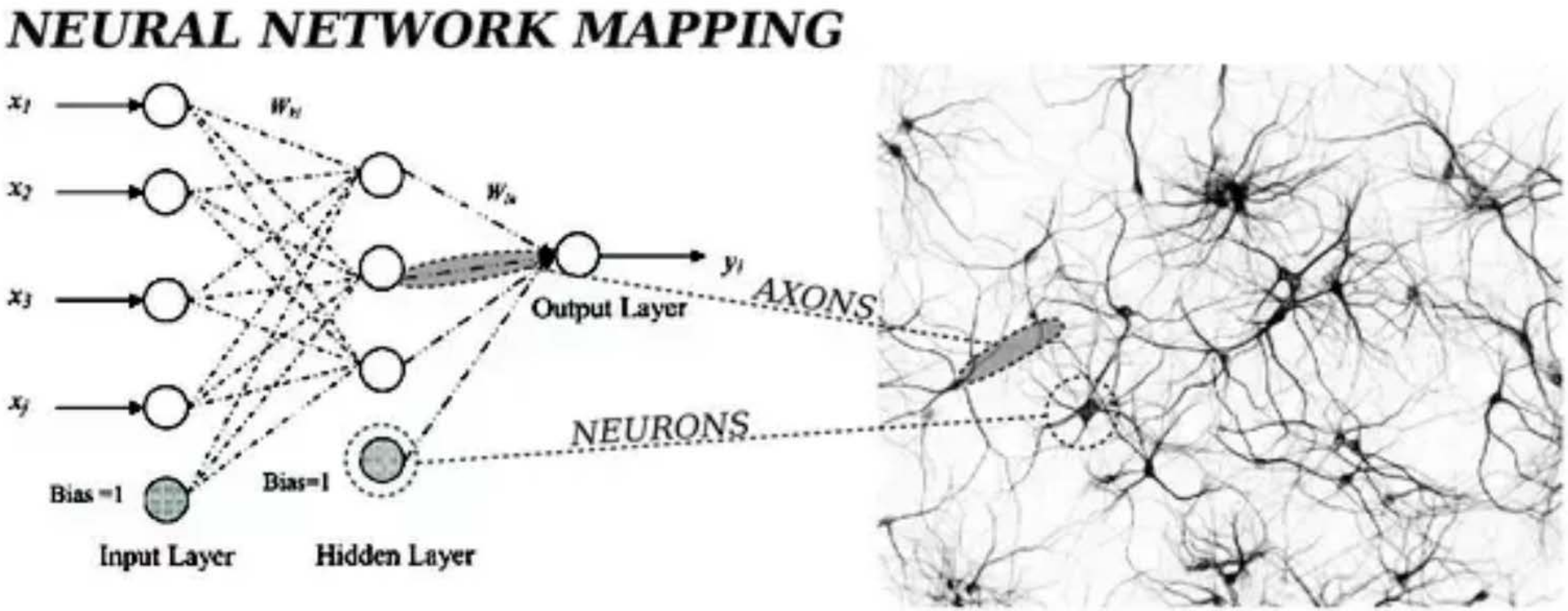
NN Introduction

- Neural network is the type of algorithm that tries to simulate how human brain works
- Was widely used in 80s and 90s
- Popularity diminished in late 90s
- Recently, neural network and deep learning became the state-of-art algorithms for many application
- Especially with the High Performance Computers (HPC) and cloud computing
- Neural networks and deep neural networks (called deep learning) has become an exciting research and application area in the last few years

NN and DL History

Early history of DL

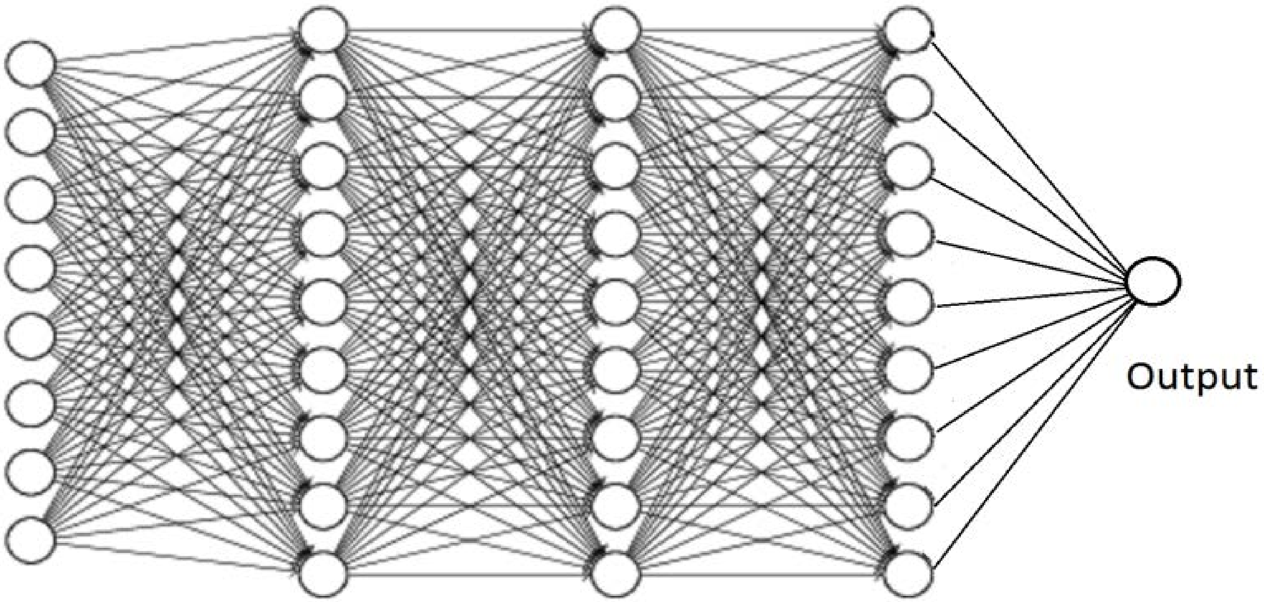




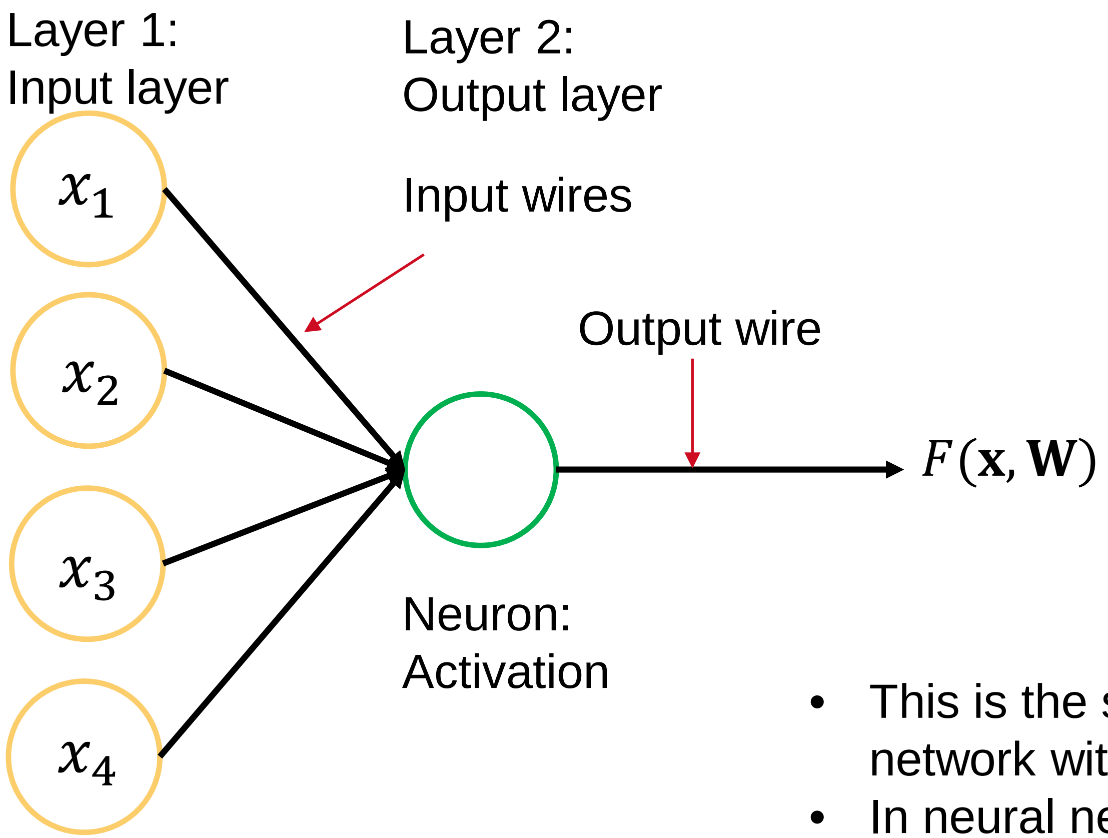
https://www.quora.com/When-will-technology-surpass-the-complexity-and-intelligence-of-the-human-brain

Neural Network Representation

A neural network is a multi-stage **regression or classification** model, typically represented by a network diagram



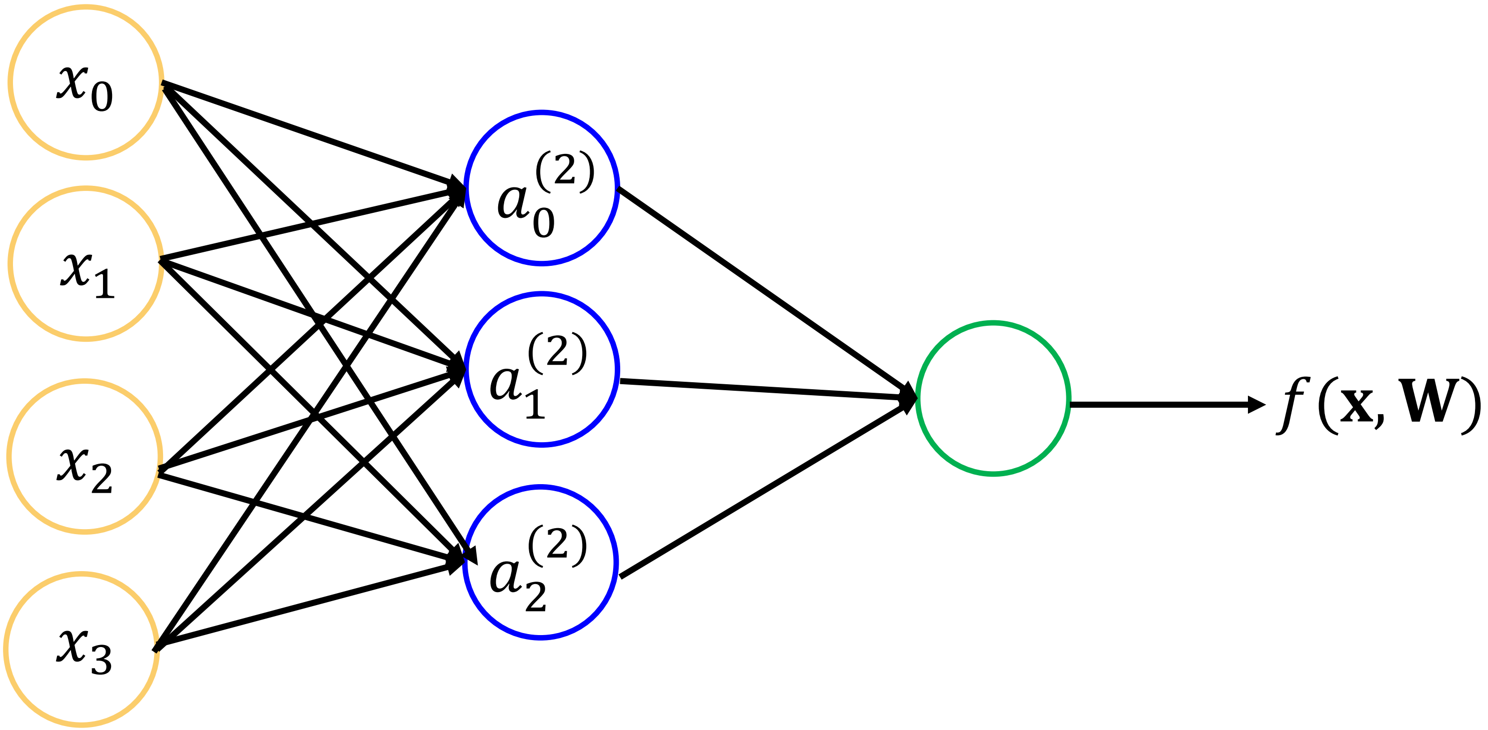
2 layer Neural Network

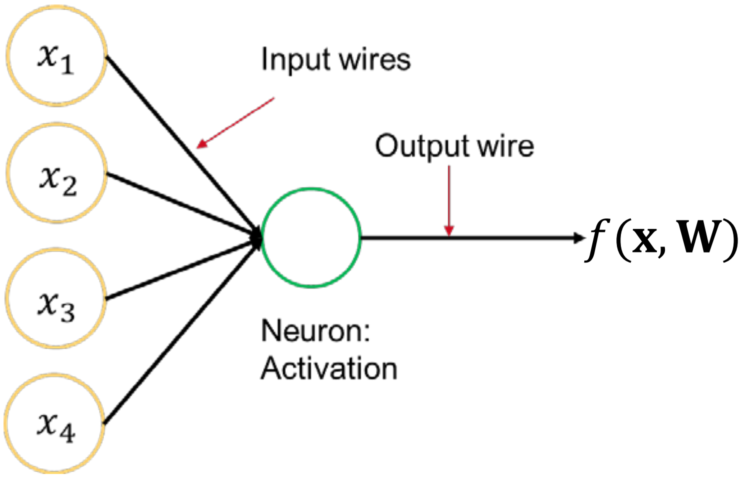


- This is the simplest neural network without hidden layer
- In neural network, parameters are also called weights

They are input, not neuron

3 layer Neural Network





The neuron is the basic processing element.

It has inputs that may come from the environment or may be the outputs of other neurons.

For a given dataset, we need to estimate weights, so that correct outputs are generated given the inputs.

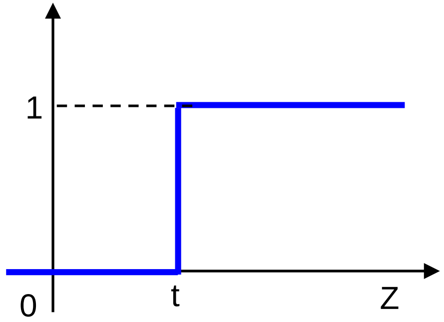
Activation Functions

- The output of a layered neural network model depends completely on the characteristics of the output layer. Units in a layer have (almost always) the same activation function.
- Typical activation functions are the sigmoid, hyperbolic tangent, and linear functions.
- The sigmoid function can only produce outputs in the range $[0,1]$, the hyperbolic tangent produces outputs in the range $[-1,1]$, and the linear function produces outputs in the range $[-\infty, \infty]$
- If we need great than 1 output in NN regression, we could use linear output units and leave the non-linear sigmoid or hyperbolical tangent units for the hidden layer. Bounded nonlinearities in the output layer are best left for cases when you require "almost binary" outputs or likelihoods (probabilities).

Activation Functions

Threshold function

$$\sigma(z) = \begin{cases} 1 & \text{if } z \geq t \\ 0 & \text{if } z < t \end{cases}$$



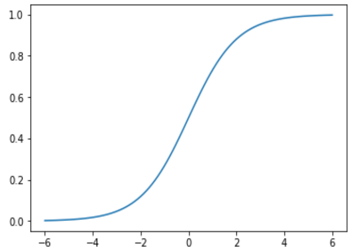
Generalized logistic (sigmoid) function

$$\sigma_{s,l}(z) = \frac{1}{1 + e^{-s(z-l)}}$$

s controls the steepness and l controls the location

Logistic (sigmoid) function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



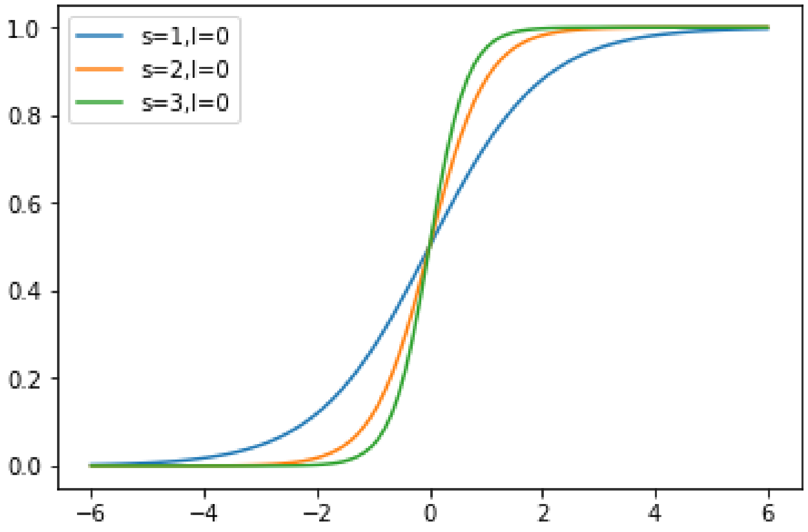
$s = 1$ and $l = 0$

Output range $(0,1)$

https://en.wikipedia.org/wiki/Activation_function

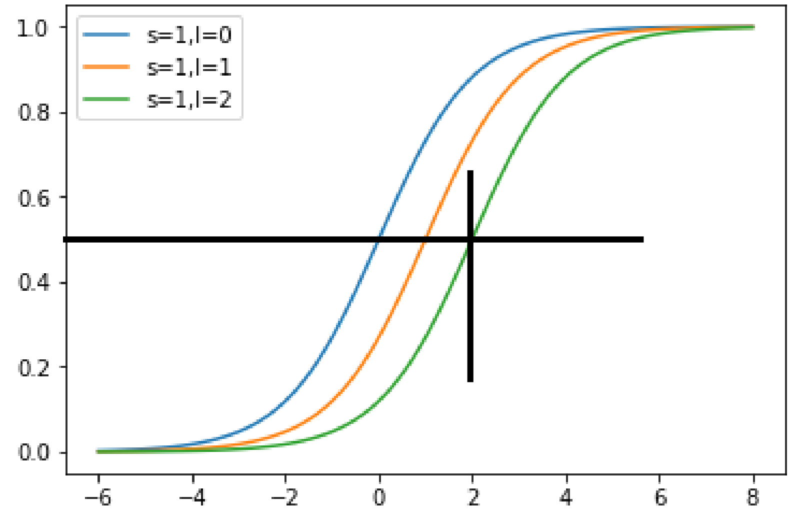
Sigmoid Function

s controls the steepness of sigmoid function. Observations?



Or we can see that s controls activation rate. larger s amounts to a shaper activation (closer to the threshold function)

l controls the location of sigmoid function



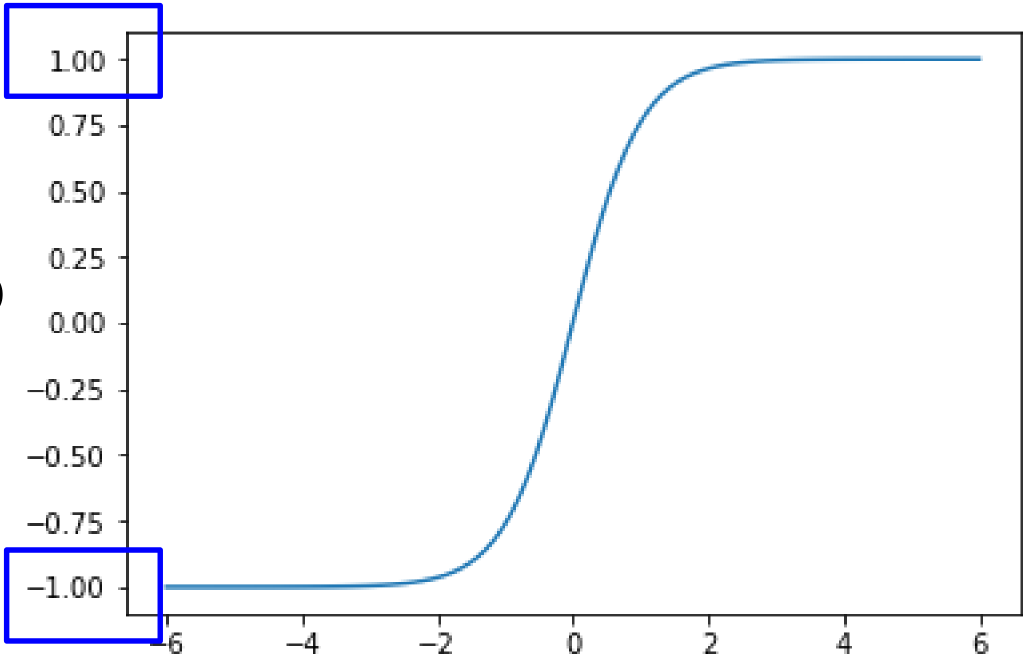
l shifts activation threshold.

Hyperbolic Tangent Function

This function is defined as the ratio of the difference and sum of two exponential functions in the points z and -z:

$$\sigma(z) = \frac{2}{1+e^{-2z}} - 1 = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

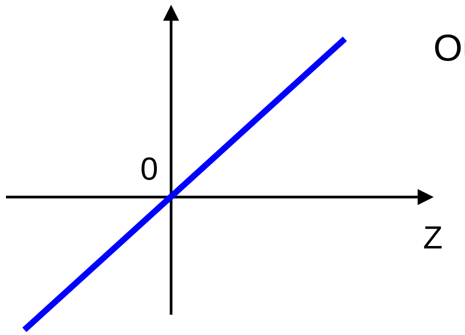
Output range $(-1,1)$



Identity Activation Function

The simplest activation function, one that is commonly used for the output layer activation function in regression problems, is the identity/linear activation function:

$$\sigma(z) = z$$



Output range $(-\infty, +\infty)$

Why use an identity activation function?

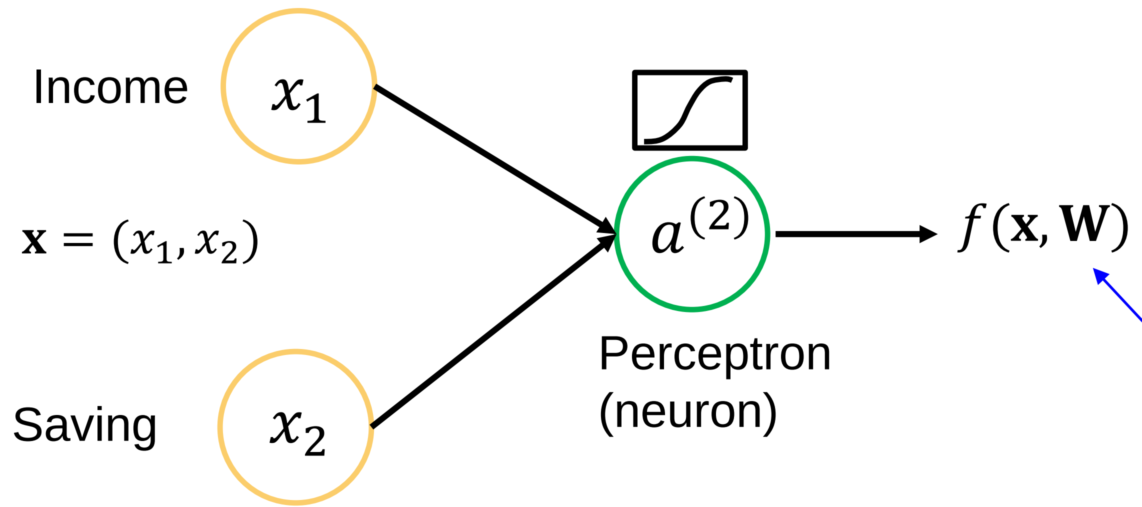
- For example, a multi-layer network that has nonlinear activation functions amongst the hidden units and an output layer that uses the identity activation function implements a powerful form of nonlinear regression. Specifically, the network can predict **continuous** target values using a linear combination of signals that arise from one or more layers of nonlinear transformations of the input.

Forward Propagation
2 Layer Neural Network

2 Layer NN

Let's start with a 2 layer NN for regression

$f(\mathbf{x}, \mathbf{W})$
is our prediction with NN



In regression, we want to make the mean square error to be minimized

$N = 3$

-Income	-Saving
4	7
1	8
3	2

$d = 2$

Feature scaling/ normalization

-expense
5
2
3

Feature scaling/ normalization

$N = 3$

-Income	-Saving
4	7
1	8
3	2

-expense
5
2
3

$d = 2$

Divide by maximum of x_1, x_2, t respectively

-Income	-Saving
1	0.875
0.25	1
0.75	0.25

-expense
1
0.4
0.6

Why scaling?

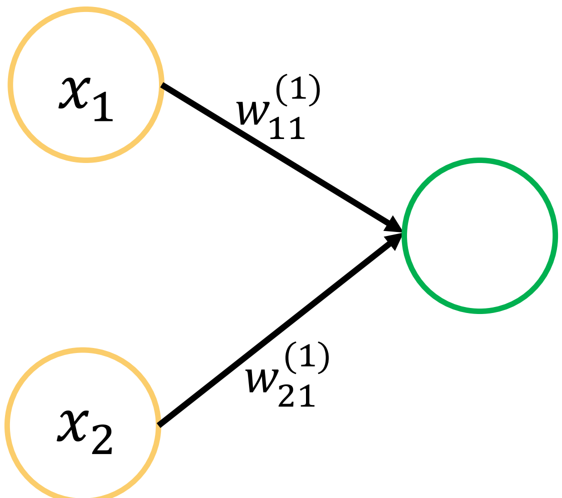
- There are a variety of practical reasons why scaling the inputs can make training faster and reduce the chances of getting stuck in local optima.

Step 1

Income

Saving

1.0	0.875
0.25	1.0
0.75	0.25



- Here we have two weights, so $\boldsymbol{W}^{(1)}$ contains two elements $w_{11}^{(1)}$ and $w_{21}^{(1)}$
- $a^{(1)}$ can also represent the input layer
- $\boldsymbol{Z}^{(2)}$ denote the total weighted sum of inputs to units in layer 2

Inputting the first data $\boldsymbol{x}_1 = (1.0, 0.875)$, we have the first $z_{11}^{(2)}$

$$1 * w_{11}^{(1)} + 0.875 * w_{21}^{(1)} = z_{11}^{2}$$

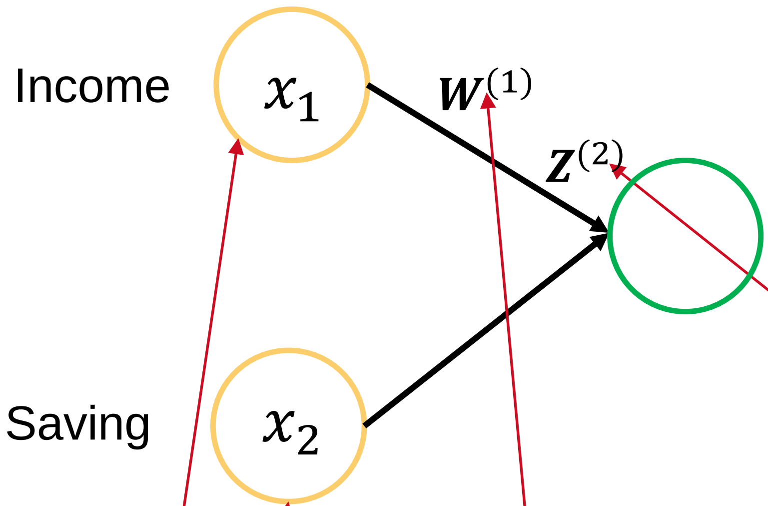
Inputting the first data $\boldsymbol{x}_2 = (0.25, 1.0)$, we have the first $z_{21}^{(2)}$

$$0.25 * w_{11}^{(1)} + 1.0 * w_{21}^{(1)} = z_{21}^{2}$$

Inputting the first data $\boldsymbol{x}_2 = (0.25, 1.0)$, we have the first $z_{21}^{(2)}$

$$0.75 * w_{11}^{(1)} + 0.25 * w_{21}^{(1)} = z_{31}^{2}$$

Step 1



- Here we have two weights, so $\boldsymbol{W}^{(1)}$ contains two elements
- $a^{(1)}$ can also represent the input layer
- $\boldsymbol{Z}^{(2)}$ denote the total weighted sum of inputs to units in layer 2

1	0.875
0.25	1
0.75	0.25

$$\times \begin{bmatrix} w_{11}^{(1)} \\ w_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} 1w_{11}^{(1)} + 0.875w_{21}^{(1)} \\ 0.25w_{11}^{(1)} + 1w_{21}^{(1)} \\ 0.75w_{11}^{(1)} + 0.25w_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} z_{11}^{(2)} \\ z_{21}^{(2)} \\ z_{31}^{(2)} \end{bmatrix}$$

$$\mathbf{X} \times \boldsymbol{W}^{(1)} = \boldsymbol{Z}^{(2)}$$

3×2 $\quad 2 \times 1$ $\quad 3 \times 1$

Step 2

$$\boldsymbol{Z}^{(2)}$$

$$a_1^{(2)} \longrightarrow f(\mathbf{x}, \mathbf{W})$$

$$\mathbf{Z}^{(2)} = \begin{bmatrix} z_{11}^{(2)} \\ z_{21}^{(2)} \\ z_{31}^{(2)} \end{bmatrix}$$

3×1 matrix (vector),
Each row represents one example
Each column represents one hidden unit, here we only have one hidden unit

$$f(\mathbf{x}, \mathbf{W}) = \mathrm{a}^{(2)} = \sigma(\mathbf{Z}^{(2)})$$

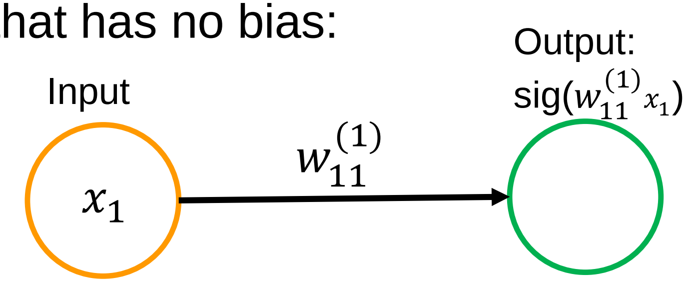
Apply activation function for **EACH** element of the matrix $\boldsymbol{Z}^{(2)}$ to produce the prediction $f(\mathbf{x}, \mathbf{W}) = a^{(2)}$ (3×1)

Loss function of regression:

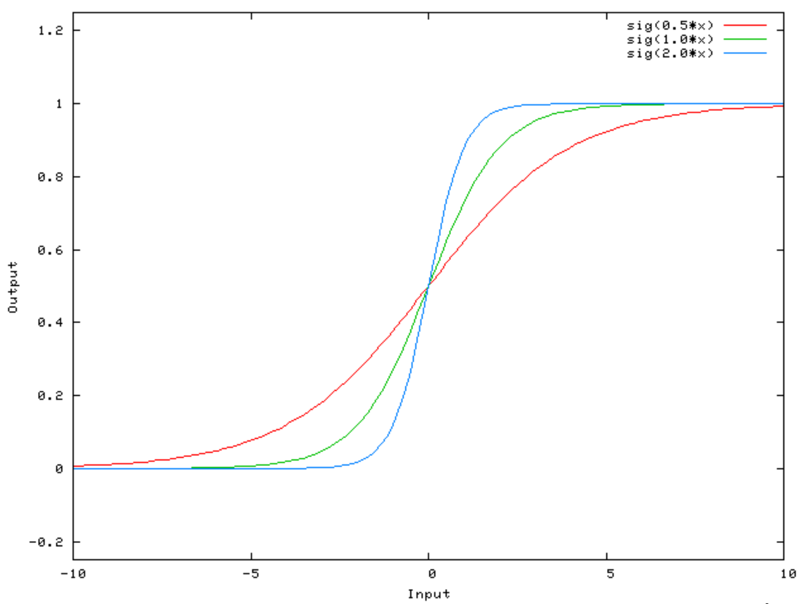
$$L(\mathbf{W}) = \frac{1}{2N} \sum_{n=1}^{N} (f(\mathbf{x}_n, \mathbf{W}) - t_n)^2$$

Bias Unit Impact

- Note that bias units don't have inputs or connections going into them, since they always output the value +1
- Biases are almost always helpful. In effect, a bias value allows you to shift the activation function to the left or right, which may be critical for successful learning.
- Consider this 1-input, 1-output network that has no bias:



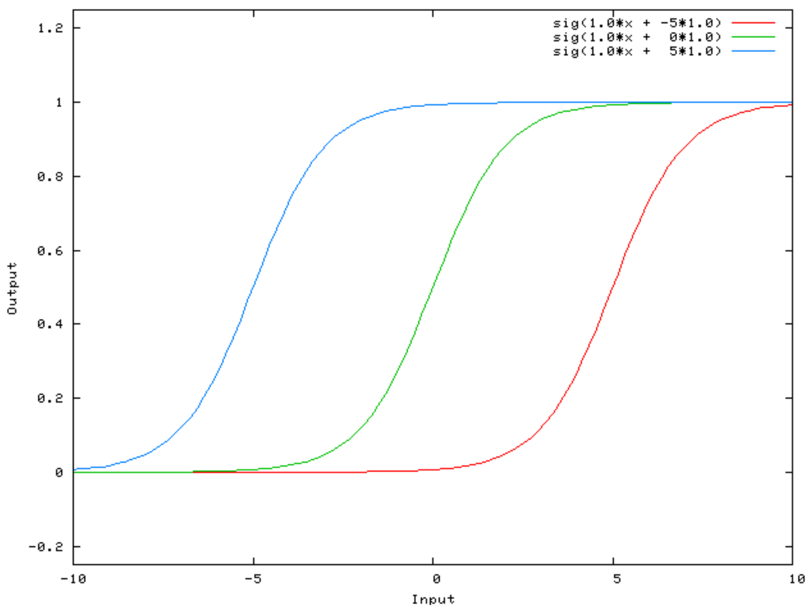
$$\sigma(z) = \frac{1}{1+e^{-z}}$$

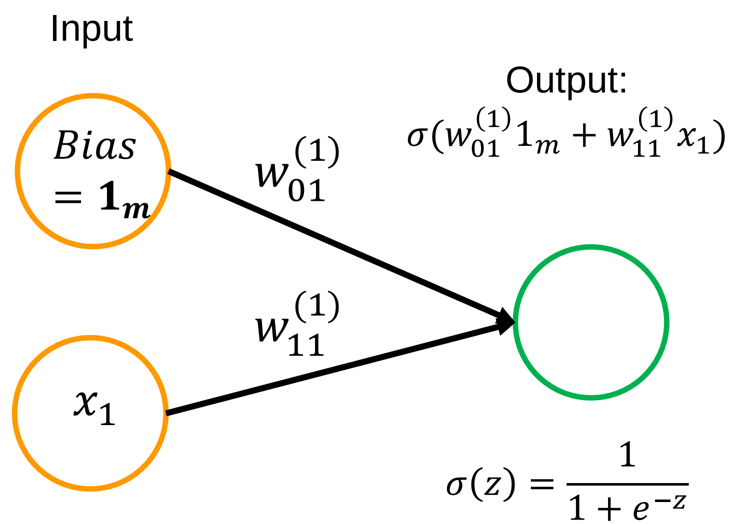


Changing the weight $w_{11}^{(1)}$ essentially changes the "steepness" of the sigmoid. That's useful, but what if you wanted the network to output 0 when x_1 is 2? Just changing the steepness of the sigmoid won't really work -- **you want to be able to shift the entire curve to the right**.

https://stackoverflow.com/questions/2480650/role-of-bias-in-neural-networks

- If we add a bias unit to this network, then the output of the network becomes $\sigma(w_{01}^{(1)} 1_m + w_{11}^{(1)} x_1)$. Here is what the output of the network looks like for various values of $w_{01}^{(1)}$:





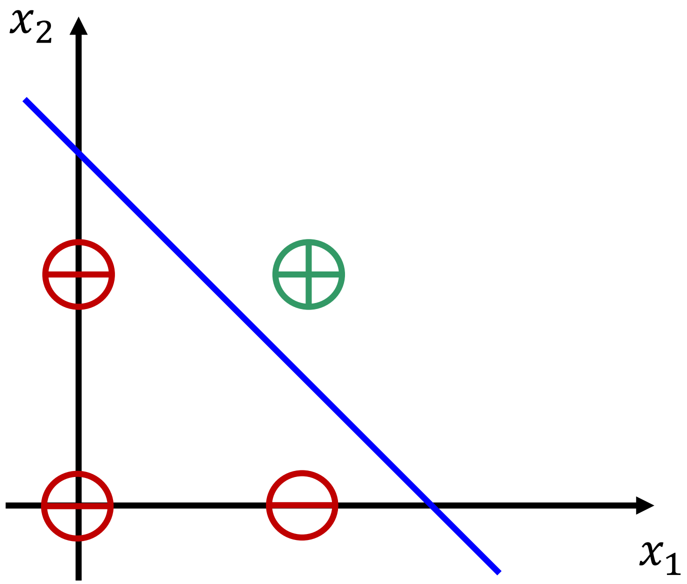
- Having a weight of -5 for $w_{01}^{(1)}$ shifts “location” of the curve to the right, which allows us to have a network that outputs 0 when x_1 is 2.
- Connect and compare this with **generalized** logistic (sigmoid) function.

Forward Propagation
2 Layer Neural Network Example

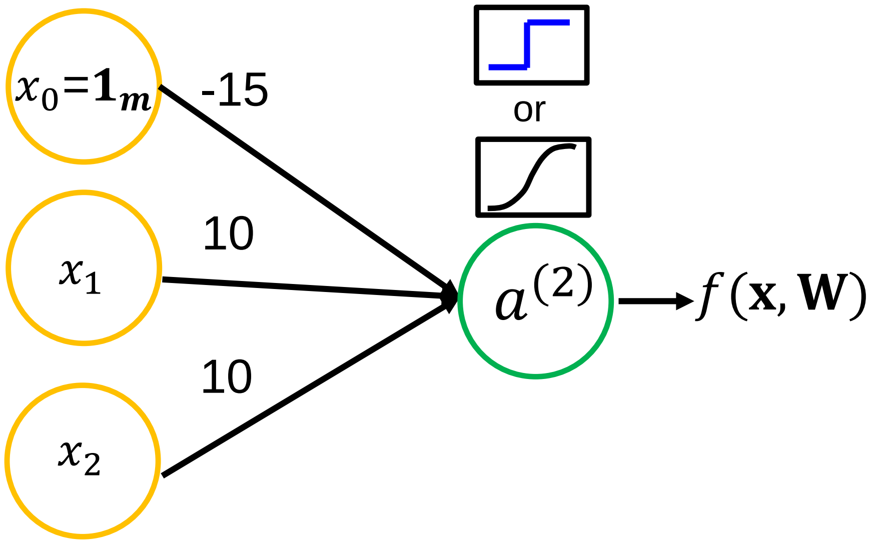
Example: AND Function

x_1	x_2	t
0	0	0 ⊖
0	1	0 ⊖
1	0	0 ⊖
1	1	1 ⊕

We want to find the blue decision boundary: **above it predict 1**; below it predict 0



Example



- Boolean functions like AND are linearly separable
- Functions like XNOR are not

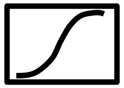
$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} w_{01}^{(1)} \\ w_{11}^{(1)} \\ w_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} 1w_{01}^{(1)} + 0w_{11}^{(1)} + 0w_{21}^{(1)} \\ 1w_{01}^{(1)} + 0w_{11}^{(1)} + 1w_{21}^{(1)} \\ 1w_{01}^{(1)} + 1w_{11}^{(1)} + 0w_{21}^{(1)} \\ 1w_{01}^{(1)} + 1w_{11}^{(1)} + 1w_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} z_{01}^{(2)} \\ z_{11}^{(2)} \\ z_{21}^{(2)} \\ z_{31}^{(2)} \end{bmatrix}$$

- Role of the intercept term/bias unit here?
- Note that bias units don't have inputs or connections going into them, since they always output the value +1

Sigmoid Activation Function

Suppose we have $\begin{bmatrix} w_{01}^{(1)} \\ w_{11}^{(1)} \\ w_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} -15 \\ 10 \\ 10 \end{bmatrix}$ How to estimate?

$$\sigma(z) = \frac{1}{1+e^{-z}}$$



1	0	0
1	0	1
1	1	0
1	1	1

$$\times \begin{bmatrix} -15 \\ 10 \\ 10 \end{bmatrix} = \begin{bmatrix} z_{01}^{(2)} \\ z_{11}^{(2)} \\ z_{21}^{(2)} \\ z_{31}^{(2)} \end{bmatrix} = \begin{bmatrix} -15 \\ -5 \\ -5 \\ 5 \end{bmatrix}$$

$$a^2 = \sigma\left(\mathbf{Z}^{(2)}\right) = \begin{bmatrix} \approx 0 \\ \approx 0 \\ \approx 0 \\ \approx 1 \end{bmatrix}$$

Predictions $f(\mathbf{x}, \mathbf{W}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

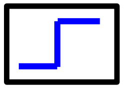
Actual observations

0
0
0
1

Threshold Activation Function

0 is the threshold

$$\sigma(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$



$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} -15 \\ 10 \\ 10 \end{bmatrix} = \begin{bmatrix} z_{01}^{(2)} \\ z_{11}^{(2)} \\ z_{21}^{(2)} \\ z_{31}^{(2)} \end{bmatrix} = \begin{bmatrix} -15 \\ -5 \\ -5 \\ 5 \end{bmatrix}$$

$$a^2 = \sigma\left(\mathbf{Z}^{(2)}\right) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

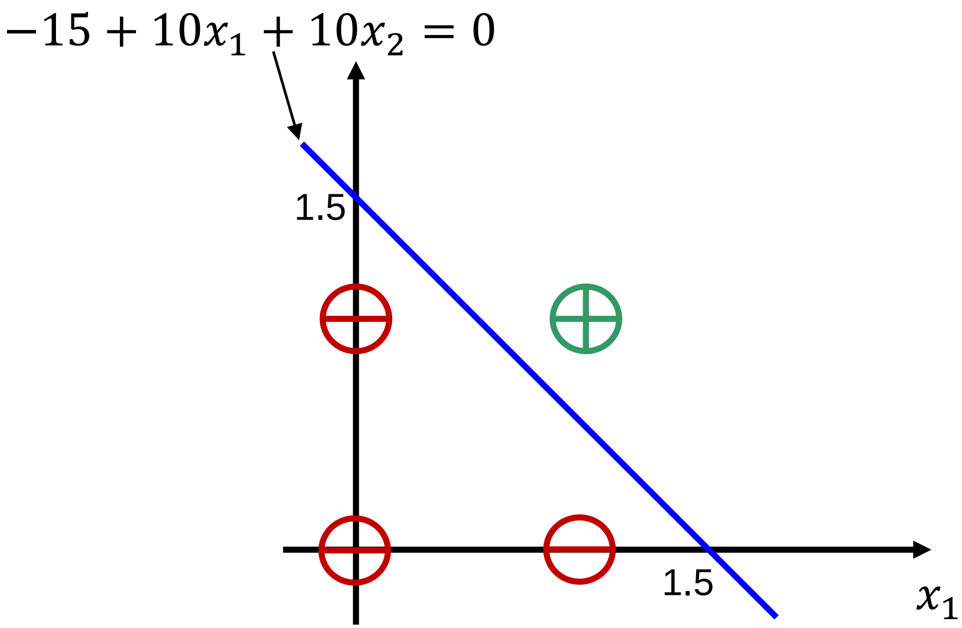
Predictions $f(\mathbf{x}, \mathbf{W}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

Actual observations

0
0
0
1

AND Function Summary

x_1	x_2	t
0	0	0
0	1	0
1	0	0
1	1	1



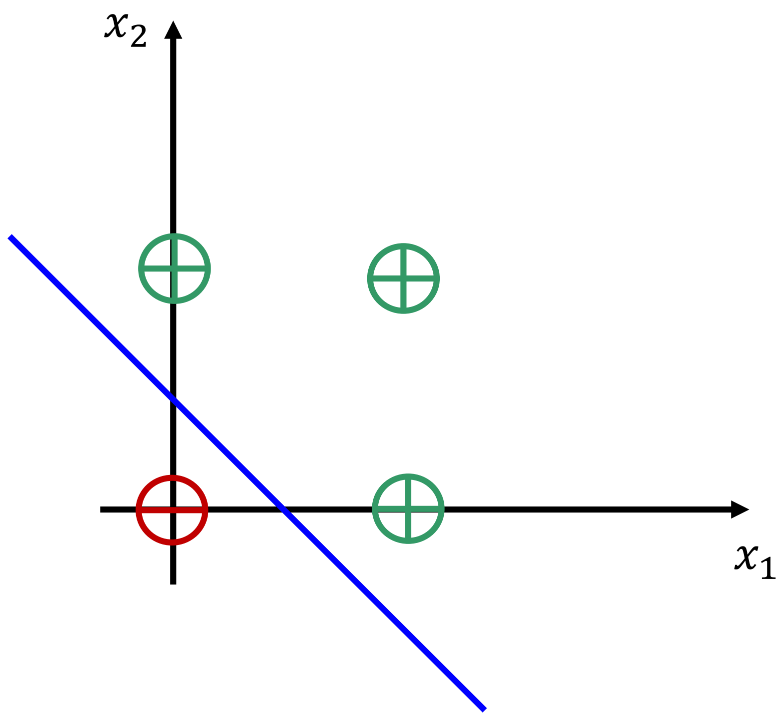
$-15 + 10x_1 + 10x_2 = 0$ is the decision boundary.
$-15 + 10x_1 + 10x_2 \geq 0$, predict 1
$-15 + 10x_1 + 10x_2 < 0$, predict 0

We will talk about how to estimate the parameters next week

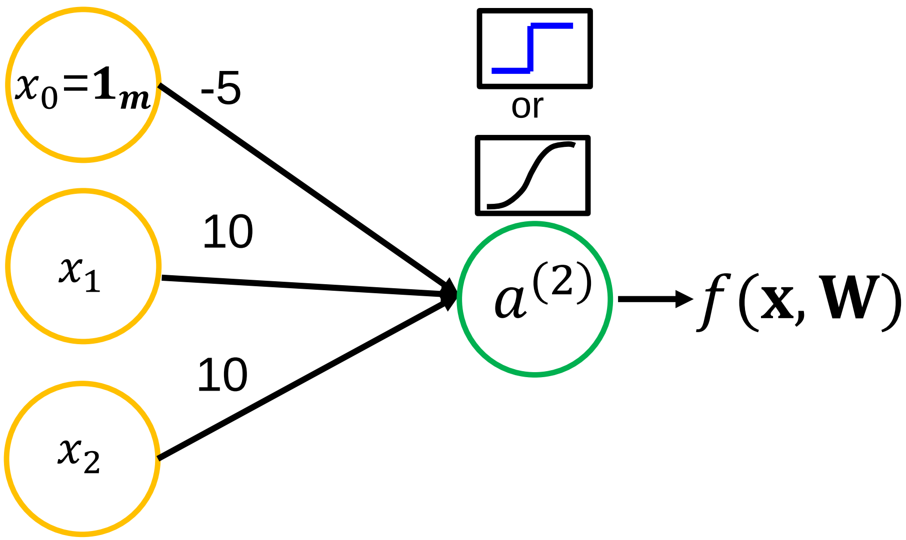
Example: OR Function

We want to find the blue decision boundary: **above it predict 1**; above it predict 0. Estimated NN?

x_1	x_2	t
0	0	0
0	1	1
1	0	1
1	1	1



Example: OR Function



$$\begin{bmatrix} w_{01}^{(1)} \\ w_{11}^{(1)} \\ w_{21}^{(1)} \end{bmatrix} = \begin{bmatrix} -5 \\ 10 \\ 10 \end{bmatrix}$$

This NN can achieve the OR function.
Why?

Forward Propagation
3 Layer Neural Network

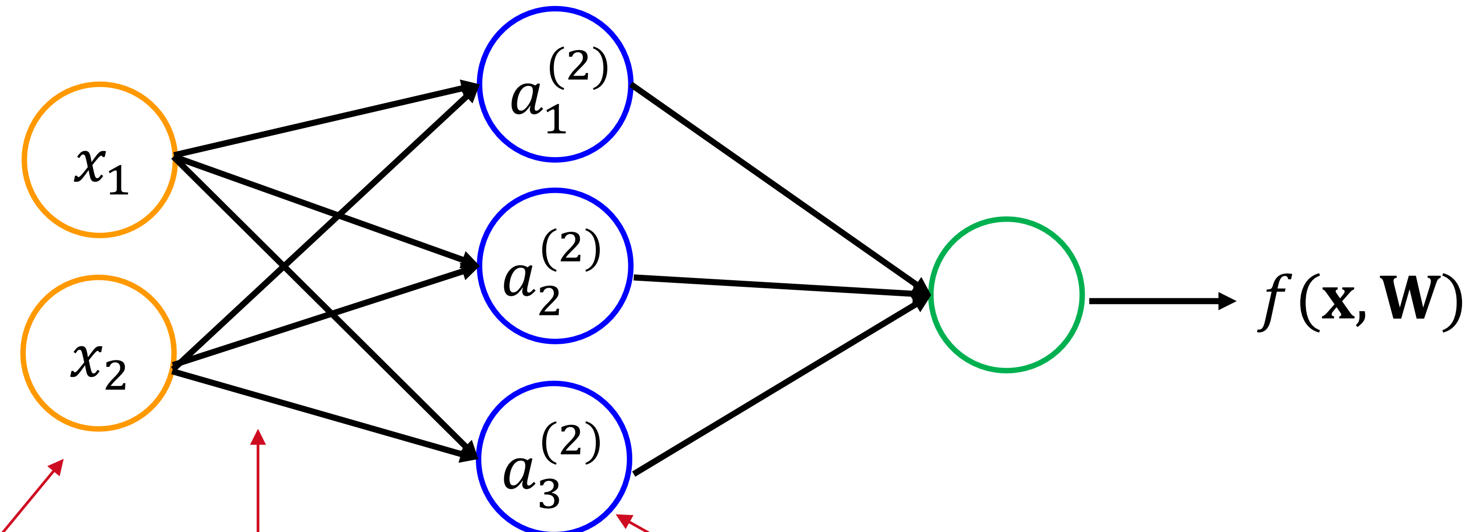
3 Layers Neural Network

Layer 1:
Input layer

Layer 2:
Hidden layer

Layer 3:
Output layer

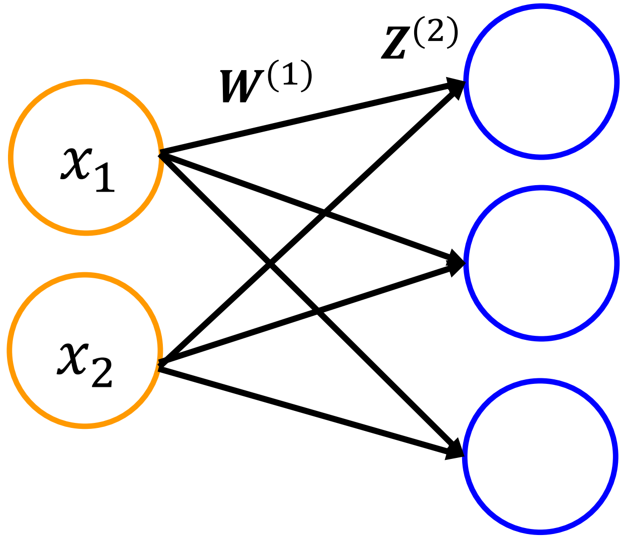
No bias unit for simplicity



1	0.875
0.25	1
0.75	0.25

$$\times \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \end{bmatrix} = \begin{bmatrix} 1w_{11}^{(1)} + 0.875w_{21}^{(1)} & 1w_{12}^{(1)} + 0.875w_{22}^{(1)} & 1w_{13}^{(1)} + 0.875w_{23}^{(1)} \\ 0.25w_{11}^{(1)} + 1w_{21}^{(1)} & 0.25w_{12}^{(1)} + 1w_{22}^{(1)} & 0.25w_{13}^{(1)} + 1w_{23}^{(1)} \\ 0.75w_{11}^{(1)} + 0.25w_{21}^{(1)} & 0.75w_{12}^{(1)} + 0.25w_{22}^{(1)} & 0.75w_{13}^{(1)} + 0.25w_{23}^{(1)} \end{bmatrix}$$

Step 1



Here we have 6 weights, so $W^{(1)}$ contains 6 elements

The nice matrix representation does both multiplication and summation jobs

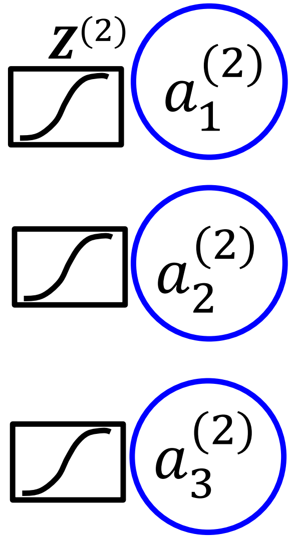
$$\begin{bmatrix} 1 & 0.875 \\ 0.25 & 1 \\ 0.75 & 0.25 \end{bmatrix} \times \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ \boxed{w_{21}^{(1)}} & w_{22}^{(1)} & w_{23}^{(1)} \end{bmatrix} = \begin{bmatrix} 1w_{11}^{(1)} + 0.875w_{21}^{(1)} & 1w_{12}^{(1)} + 0.875w_{22}^{(1)} & \boxed{1w_{13}^{(1)} + 0.875w_{23}^{(1)}} \\ 0.25w_{11}^{(1)} + 1w_{21}^{(1)} & 0.25w_{12}^{(1)} + 1w_{22}^{(1)} & 0.25w_{13}^{(1)} + 1w_{23}^{(1)} \\ 0.75w_{11}^{(1)} + 0.25w_{21}^{(1)} & 0.75w_{12}^{(1)} + 0.25w_{22}^{(1)} & 0.75w_{13}^{(1)} + 0.25w_{23}^{(1)} \end{bmatrix}$$

$$\mathbf{X} \quad \times \quad W^{(1)} \quad = \quad Z^{(2)}$$

Weights (connection) between unit 2 (x_2) in input layer (layer 1) to unit 1 $a_1^{(2)}$ in hidden layer (layer 2)

3 × 3 matrix,
Each row represents one example
Each column represents one hidden unit

Step 2



- $a^{(1)}$ can also represent the input layer
- $\boldsymbol{Z}^{(2)}$ denote the total weighted sum of inputs to units in layer 2

$$\mathbf{Z}^{(2)} = \begin{bmatrix} z_{11}^{(2)} & z_{12}^{(2)} & z_{13}^{(2)} \\ z_{21}^{(2)} & z_{22}^{(2)} & z_{23}^{(2)} \\ z_{31}^{(2)} & z_{32}^{(2)} & z_{33}^{(2)} \end{bmatrix}$$

Apply activation function for **EACH** element of the $\boldsymbol{Z}^{(2)}$ matrix

$$a^{(2)} = \sigma(\mathbf{Z}^{(2)})$$

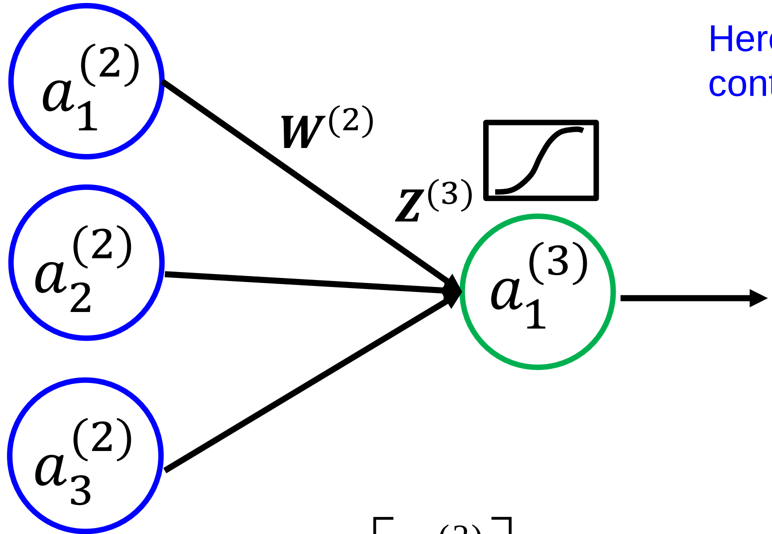
3 by 3 matrix

Now generate the output

Step 3&4

Here we have 3 weights, so $\boldsymbol{W}^{(2)}$ contains 3 elements

Step 3



$f(\mathbf{x}, \mathbf{W}) = a^{(3)} = \sigma\left(\boldsymbol{Z}^{(3)}\right)$

$a^{(2)}$

3×3

$$\mathbf{W}^{(2)} = \begin{bmatrix} w_{11}^{(2)} \\ w_{21}^{(2)} \\ w_{31}^{(2)} \end{bmatrix}$$

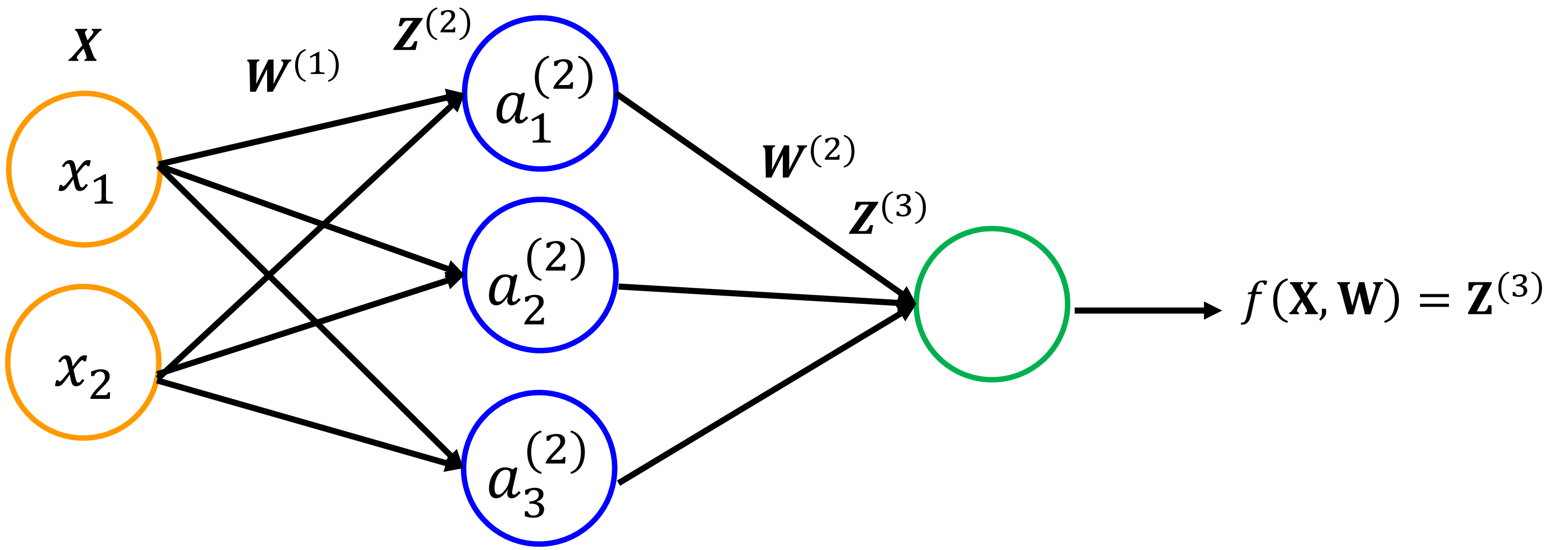
3×1

$\mathbf{Z}^{(3)} = a^{(2)}\mathbf{W}^{(2)}$ 3×1

Step 4 $f(\mathbf{x}, \mathbf{W}) = a^{(3)} = \sigma\left(\boldsymbol{Z}^{(3)}\right)$

Apply activation function on $\boldsymbol{Z}^{(3)}$, and generate final output/prediction $f(\mathbf{x}, \mathbf{W}) = a^{(3)}$

Sum it up- no bias unit



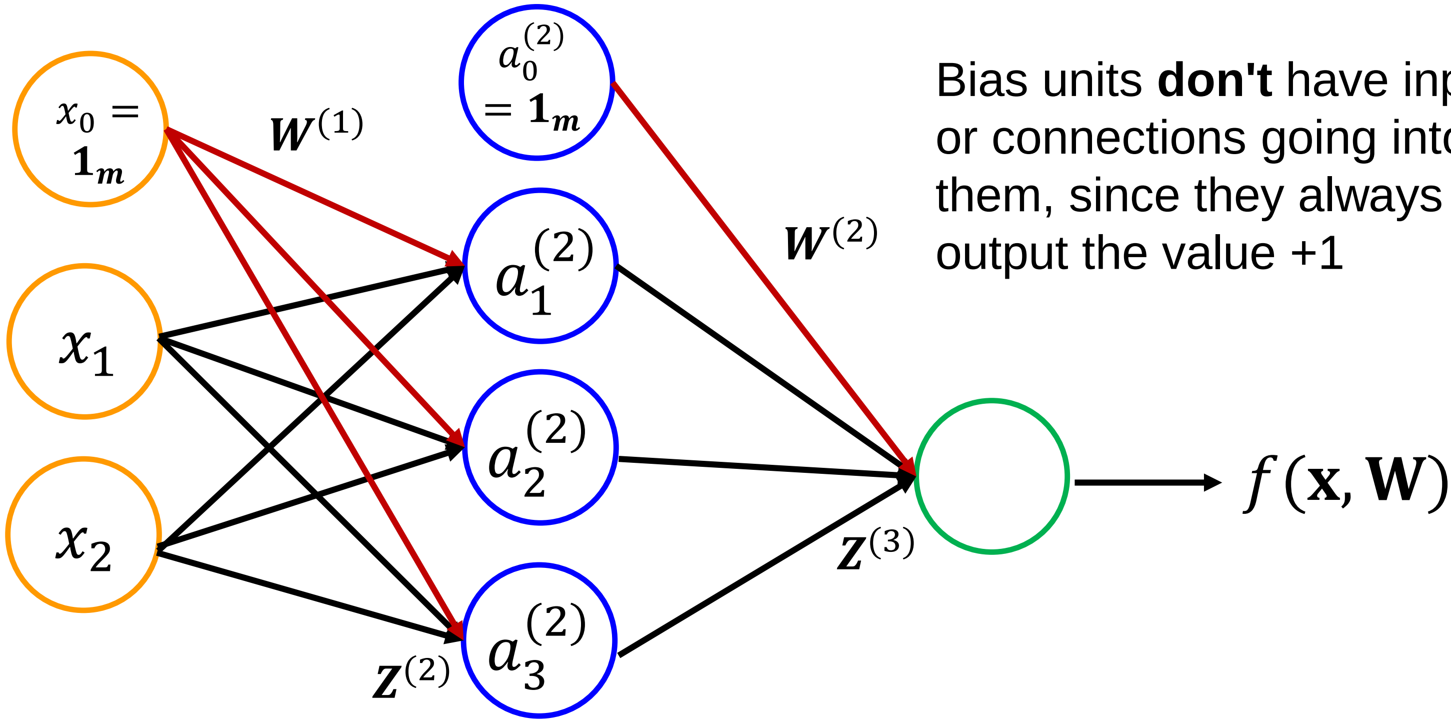
$\boldsymbol{Z}^{(2)} = \mathbf{X}\mathbf{W}^{(1)}$ $a^{(2)} = \sigma(\boldsymbol{Z}^{(2)})$ $\boldsymbol{Z}^{(3)} = a^{(2)}\mathbf{W}^{(2)}$ $f(\mathbf{x}, \mathbf{W}) = a^{(3)} = \sigma(\boldsymbol{Z}^{(3)})$

Activation

Activation

- $a_i^{(j)}$: “activation” of unit i in layer j
- $\mathbf{W}^{(j)}$: weight matrix that controls function mapping from layer j to layer $j+1$
- $\sigma()$ is the activation function
- No bias unit

Sum it up- with bias unit



Bias units **don't** have inputs or connections going into them, since they always output the value +1

$$Z^{(2)} = \mathbf{X}\mathbf{W}^{(1)} \qquad a^{(2)} = \sigma(Z^{(2)}) \qquad Z^{(3)} = a^{(2)}\mathbf{W}^{(2)} \qquad f(\mathbf{x}, \mathbf{W}) = a^{(3)} = \sigma\left(Z^{(3)}\right)$$

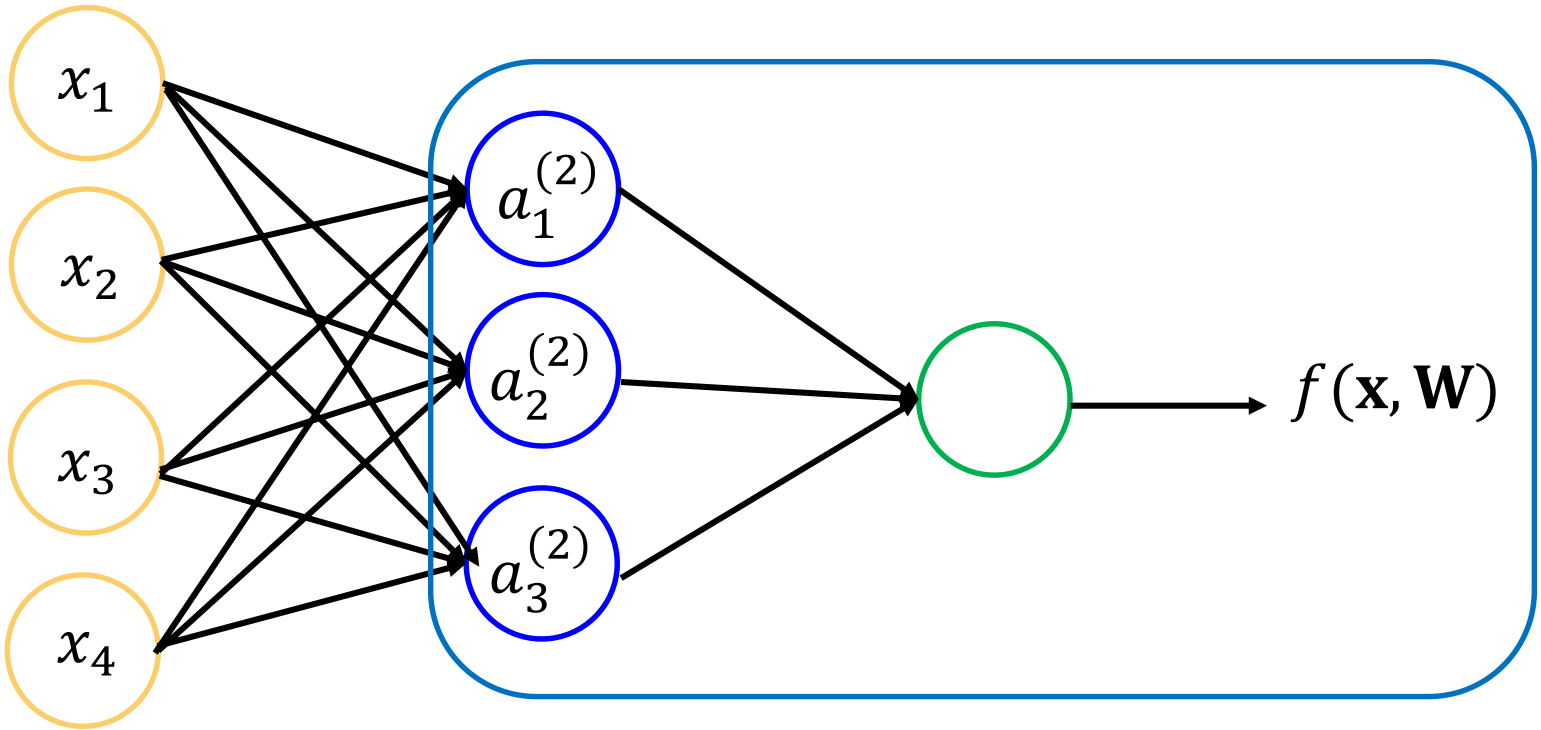
Bias unit weights

$$\mathbf{W}^{(1)} = \begin{bmatrix} \boxed{w_{01}^{(1)} \quad w_{02}^{(1)} \quad w_{03}^{(1)}} \\ w_{11}^{(1)} \quad w_{12}^{(1)} \quad w_{13}^{(1)} \\ w_{21}^{(1)} \quad w_{22}^{(1)} \quad w_{23}^{(1)} \end{bmatrix}$$

Activation

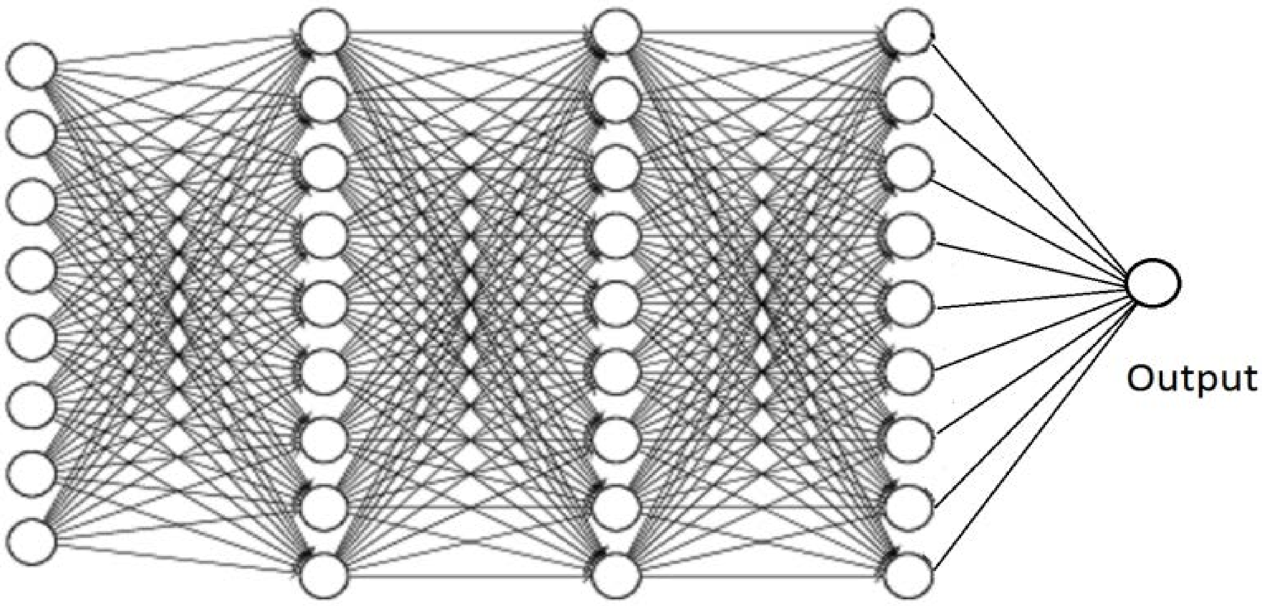
$$\mathbf{W}^{(2)} = \begin{bmatrix} \boxed{w_{01}^{(2)}} \\ w_{11}^{(2)} \\ w_{21}^{(2)} \\ w_{31}^{(2)} \end{bmatrix}$$

Activation



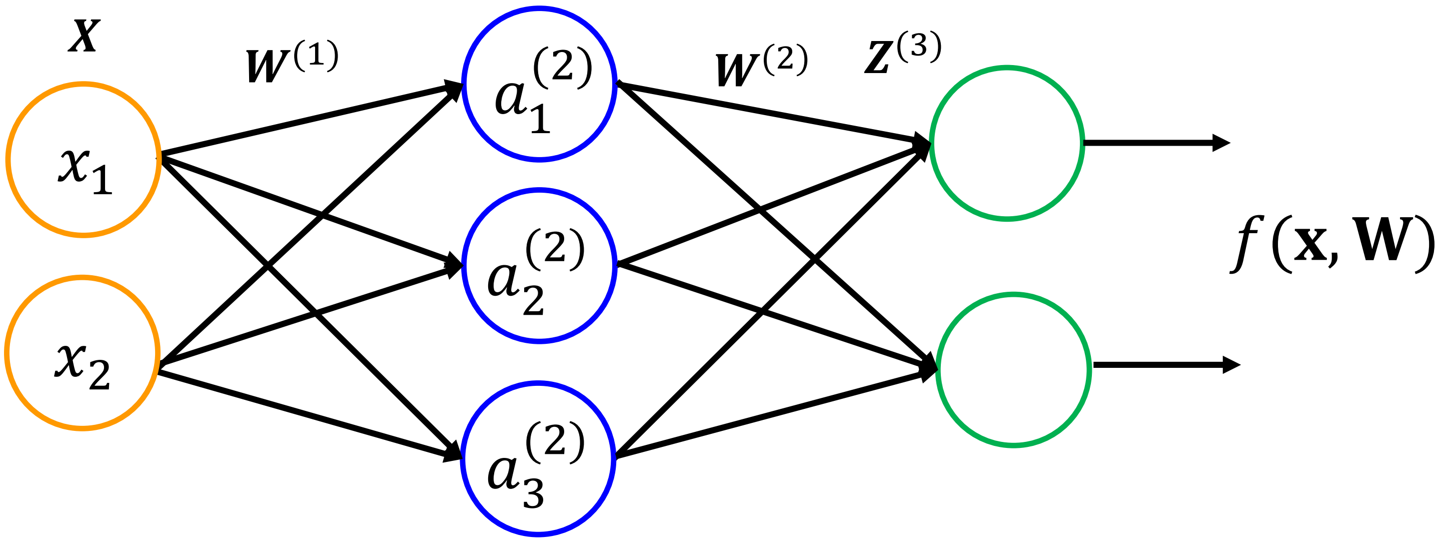
NN is learning its own features $a_1^{(2)}, a_2^{(2)}, a_3^{(2)}$

Architectures of the neural networks: patterns of connectivity between neurons.



Multiple output units

Neural networks can also have multiple output units.

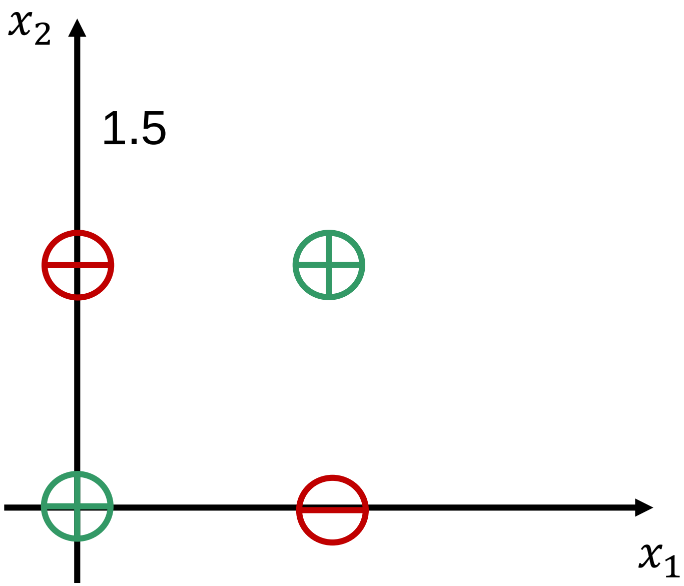


For example, in a medical diagnosis application, the vector **x** might give the input features of a patient, and the different outputs might indicate presence or absence of different diseases.

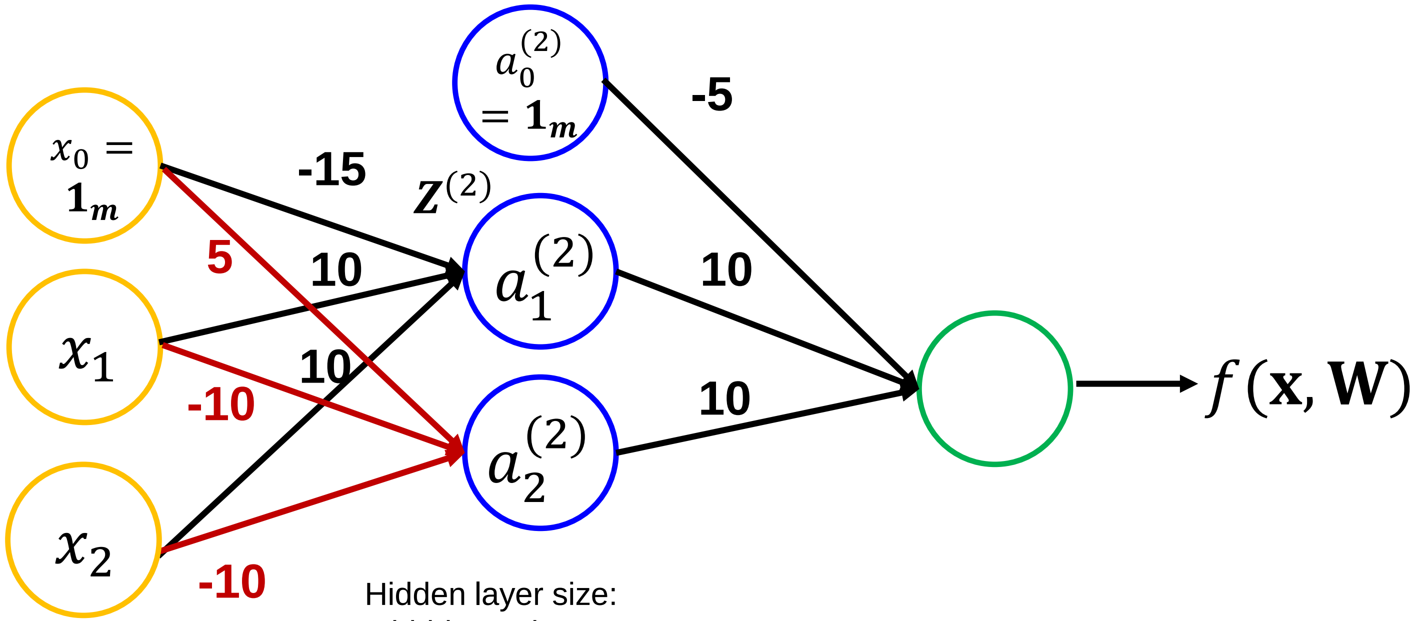
Forward Propagation
3 Layer Neural Network Example

XNOR Function

x_1	x_2	t
0	0	1
0	1	0
1	0	0
1	1	1



The problem is not linearly separable, or we cannot draw one decision boundary to separate the data

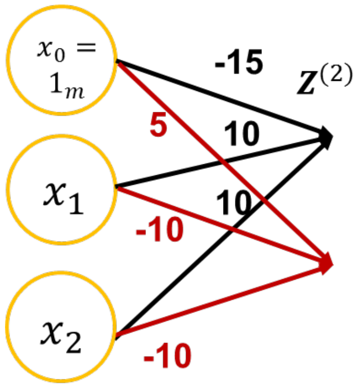


Hidden layer size:
2 hidden units

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \times \overbrace{\begin{bmatrix} w_{01}^{(1)} & w_{02}^{(1)} \\ w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix}} = \begin{bmatrix} 1w_{01}^{(1)} + 0w_{11}^{(1)} + 0w_{21}^{(1)} & 1w_{02}^{(1)} + 0w_{12}^{(1)} + 0w_{22}^{(1)} \\ 1w_{01}^{(1)} + 0w_{11}^{(1)} + 1w_{21}^{(1)} & 1w_{02}^{(1)} + 0w_{12}^{(1)} + 1w_{22}^{(1)} \\ 1w_{01}^{(1)} + 1w_{11}^{(1)} + 0w_{21}^{(1)} & 1w_{02}^{(1)} + 1w_{12}^{(1)} + 0w_{22}^{(1)} \\ 1w_{01}^{(1)} + 1w_{11}^{(1)} + 1w_{21}^{(1)} & 1w_{02}^{(1)} + 1w_{12}^{(1)} + 1w_{22}^{(1)} \end{bmatrix} = \begin{bmatrix} z_{01}^{(2)} & z_{02}^{(2)} \\ z_{11}^{(2)} & z_{12}^{(2)} \\ z_{21}^{(2)} & z_{22}^{(2)} \\ z_{31}^{(2)} & z_{32}^{(2)} \end{bmatrix}$$

$$\mathbf{X} \times \boldsymbol{W}^{(1)} = \boldsymbol{Z}^{(2)}$$

Step 1



Suppose we have

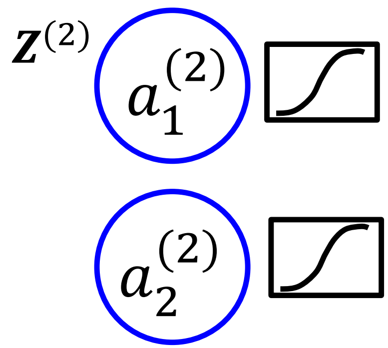
$$\begin{bmatrix} w_{01}^{(1)} & w_{02}^{(1)} \\ w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix} = \begin{bmatrix} -15 & 5 \\ 10 & -10 \\ 10 & -10 \end{bmatrix}$$

1	0	0
1	0	1
1	1	0
1	1	1

$$\times \begin{bmatrix} -15 & 5 \\ 10 & -10 \\ 10 & -10 \end{bmatrix} = \begin{bmatrix} z_{01}^{(2)} & z_{02}^{(2)} \\ z_{11}^{(2)} & z_{12}^{(2)} \\ z_{21}^{(2)} & z_{22}^{(2)} \\ z_{31}^{(2)} & z_{32}^{(2)} \end{bmatrix} = \begin{bmatrix} -15 & 5 \\ -5 & -5 \\ -5 & -5 \\ 5 & -15 \end{bmatrix}$$

Step 2

Threshold activation function with threshold at 0 is another common choice.



$$\begin{bmatrix} z_{01}^{(2)} & z_{02}^{(2)} \\ z_{11}^{(2)} & z_{12}^{(2)} \\ z_{21}^{(2)} & z_{22}^{(2)} \\ z_{31}^{(2)} & z_{32}^{(2)} \end{bmatrix} = \begin{bmatrix} -15 & 5 \\ -5 & -5 \\ -5 & -5 \\ 5 & -15 \end{bmatrix}$$

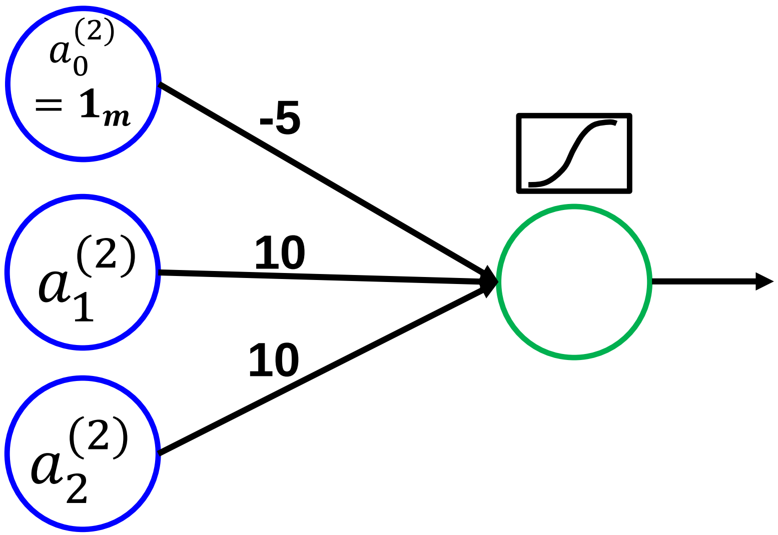
Apply activation function for EACH element of the matrix

$$a^{(2)} = \sigma\left(\mathbf{Z}^{(2)}\right) = \begin{bmatrix} \approx 0 & \boxed{\approx 1} \\ \approx 0 & \approx 0 \\ \approx 0 & \approx 0 \\ \boxed{\approx 1} & \approx 0 \end{bmatrix}$$

4 by 2 matrix

Step 3&4

Add the bias unit



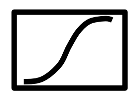
$f(\mathbf{x}, \mathbf{W}) = \sigma\left(\boldsymbol{Z}^{(3)}\right) = a^{(3)}$

Suppose:

$$a^{(2)} = \sigma\left(\boldsymbol{Z}^{(2)}\right) = \begin{bmatrix} 1 & \approx 0 & \approx 1 \\ 1 & \approx 0 & \approx 0 \\ 1 & \approx 0 & \approx 0 \\ 1 & \approx 1 & \approx 0 \end{bmatrix} \quad \mathbf{W}^{(2)} = \begin{bmatrix} w_{01}^{(2)} \\ w_{11}^{(2)} \\ w_{21}^{(2)} \end{bmatrix} = \begin{bmatrix} -5 \\ 10 \\ 10 \end{bmatrix} \quad \mathbf{Z}^{(3)} = a^{(2)}\mathbf{W}^{(2)}$$

$4 \times (2+1)$ $\quad$ $(2+1) \times 1$ $\quad$ 4×1

$$a^{(2)} = \begin{bmatrix} 1 & \approx 0 & \approx 1 \\ 1 & \approx 0 & \approx 0 \\ 1 & \approx 0 & \approx 0 \\ 1 & \approx 1 & \approx 0 \end{bmatrix} \qquad \mathbf{W}^{(2)} = \begin{bmatrix} w_{01}^{(2)} \\ w_{11}^{(2)} \\ w_{21}^{(2)} \end{bmatrix} = \begin{bmatrix} -5 \\ 10 \\ 10 \end{bmatrix}$$

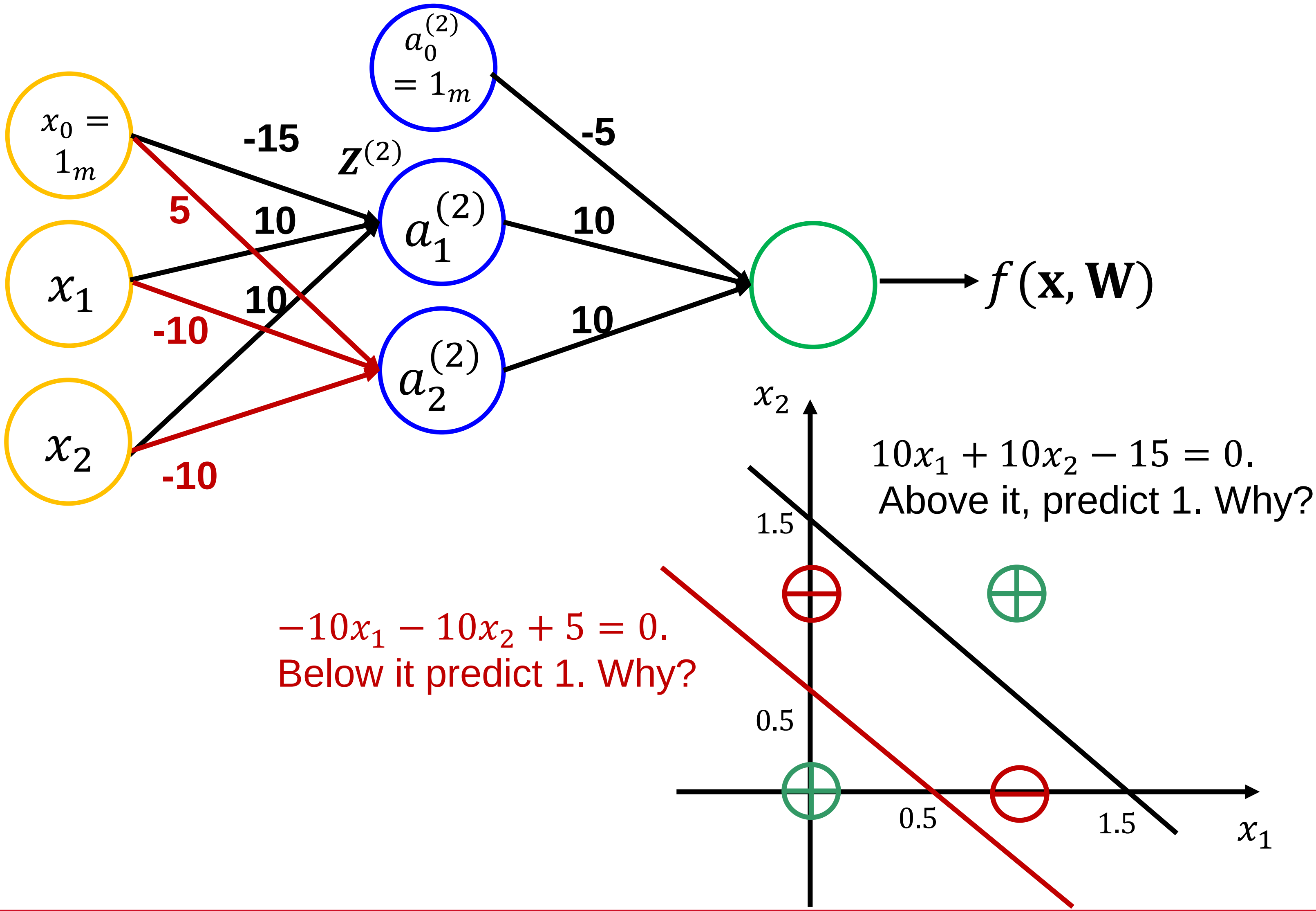


$$\mathbf{Z}^{(3)} = a^{(2)}\mathbf{W}^{(2)} = \begin{bmatrix} \approx 5 \\ \approx -5 \\ \approx -5 \\ \approx 5 \end{bmatrix} \qquad f(\mathbf{x}, \mathbf{W}) = a^{(3)} = \sigma\left(\boldsymbol{Z}^{(3)}\right) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

1
0
0
1

Try to use threshold activation function with threshold at 0.

Visualization



Derivative Exercise

- Self study/review on derivative summation rule
- Self study/review derivative chain rule
- Calculate the derivative of the following sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\sigma'(z) = ?$$